Pandas

Pandas

- A powerful Python library for data manipulation and analysis.
- **Use Cases:** Data cleaning, transformation, and analysis.
- **Data Structures:** Series and DataFrame

Key features of Pandas

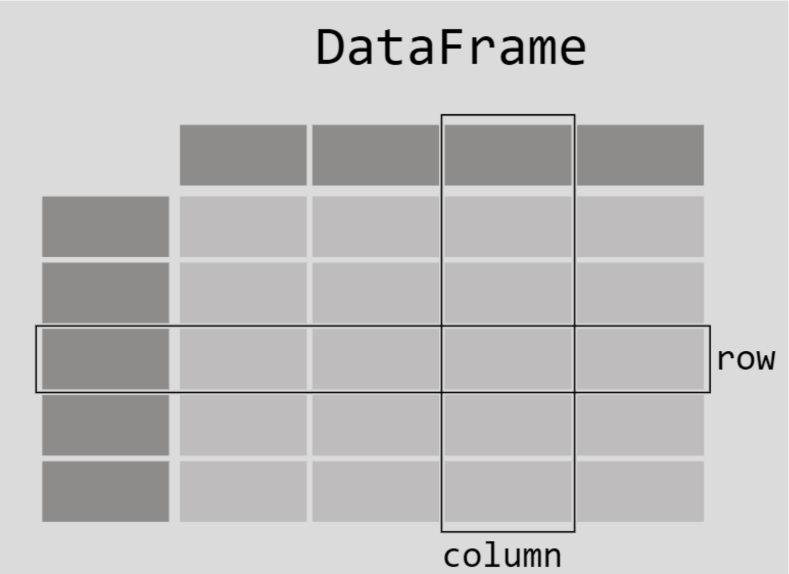
- **DataFrames and Series:** Flexible and powerful data structures.
- **Data Alignment and Missing Data Handling:** Tools for dealing with real-world data.
- **Label-Based Indexing:** Easy data selection and manipulation.
- **Group By:** Aggregation and transformation of data.

Basic data structures in Pandas

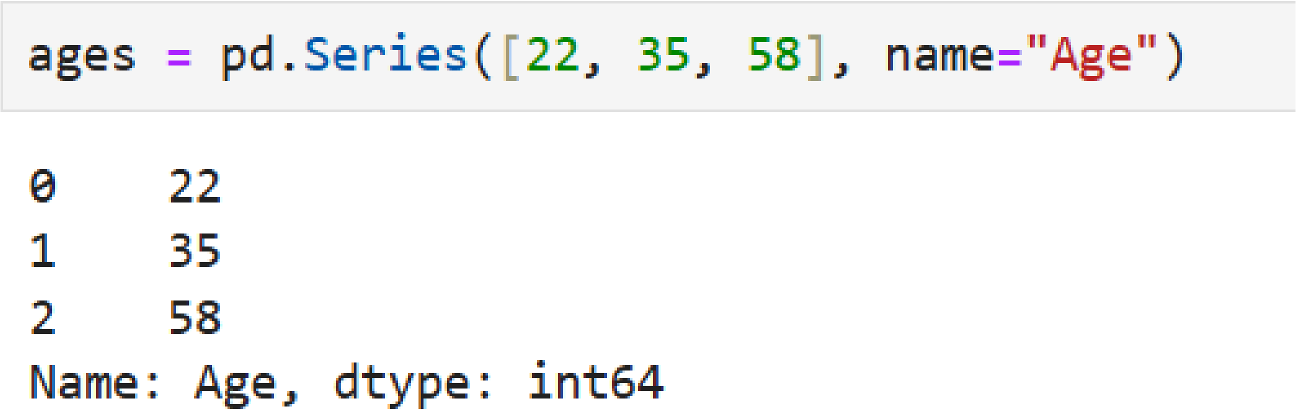
- **Series: a one-dimensional labeled array holding data of any type** such as integers, strings, Python objects etc.

- **DataFrame**: a two-dimensional data structure that holds data like a two-dimension array or a table with rows and columns.

- Each column in a dataframe is Series



Series creation

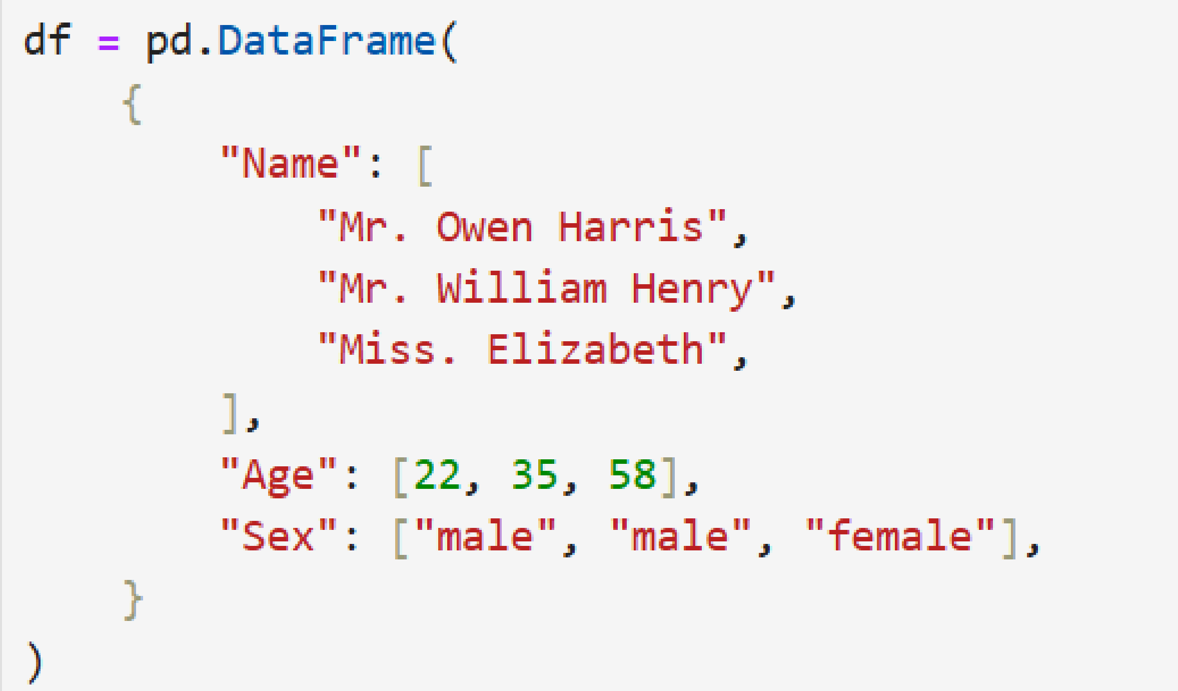


```
ages = pd.Series([22, 35, 58], name="Age")
```

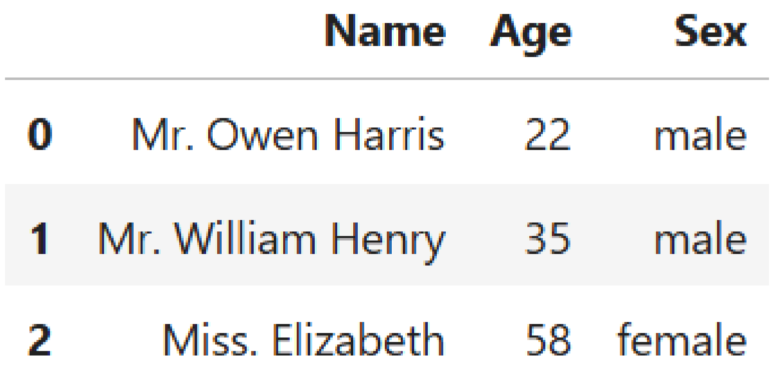
```
0    22
1    35
2    58
Name: Age, dtype: int64
```

Dataframe Creation

- Using a dictionary



```
df = pd.DataFrame(
    {
        "Name": [
            "Mr. Owen Harris",
            "Mr. William Henry",
            "Miss. Elizabeth",
        ],
        "Age": [22, 35, 58],
        "Sex": ["male", "male", "female"],
    }
)
```

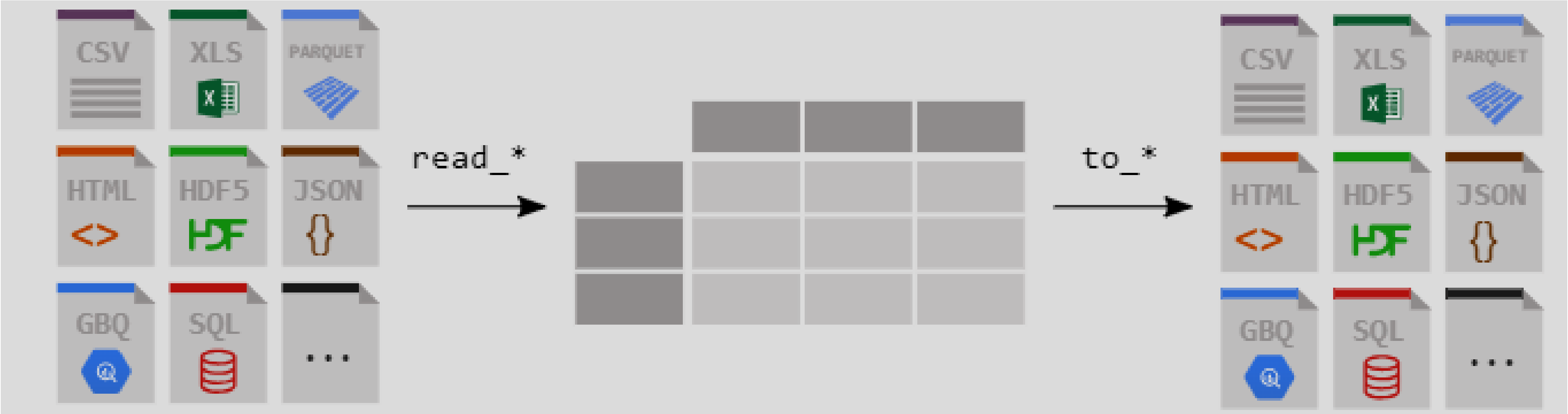


	Name	Age	Sex
0	Mr. Owen Harris	22	male
1	Mr. William Henry	35	male
2	Miss. Elizabeth	58	female

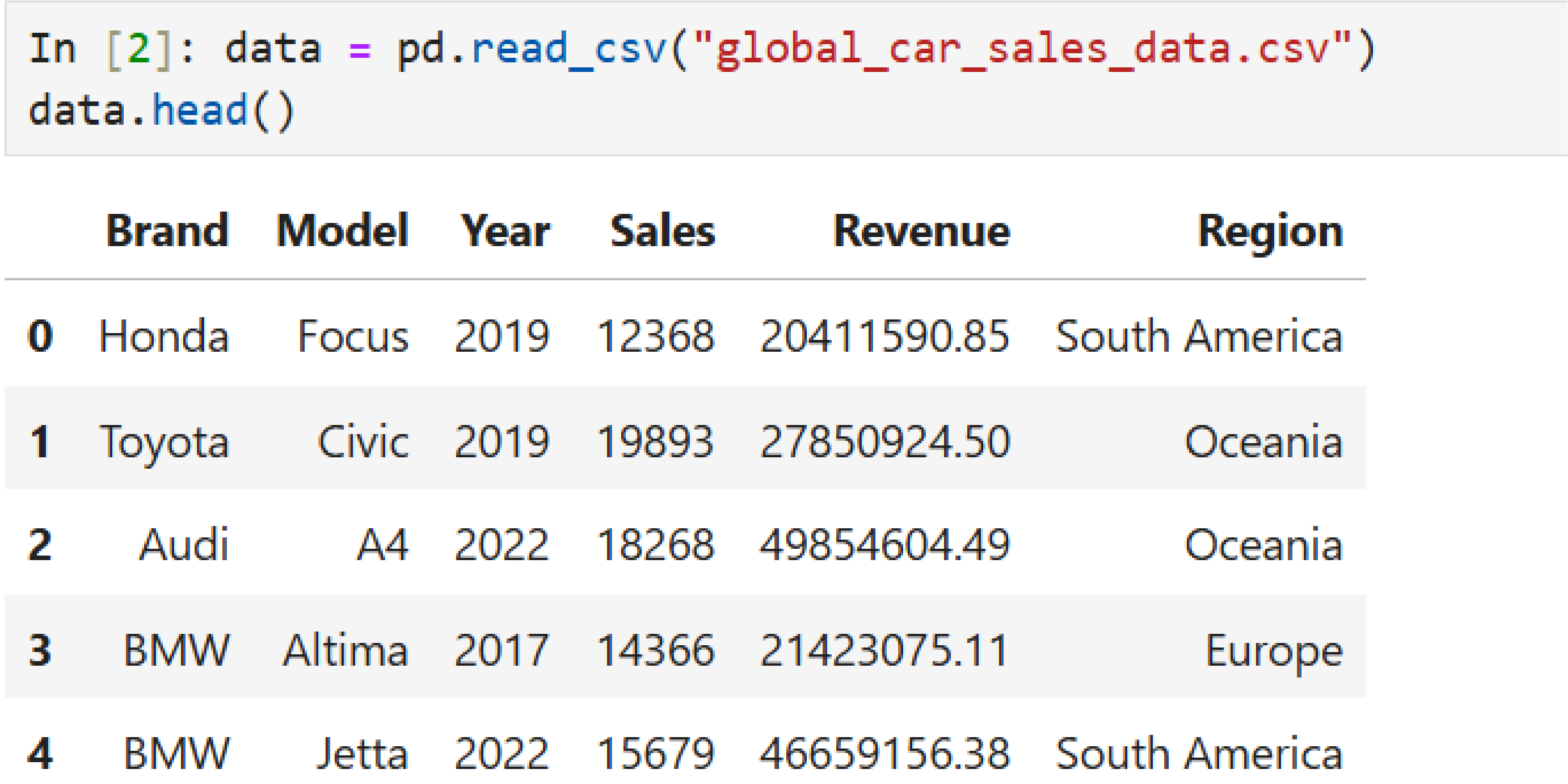
- Keys become column headers and values become column values

Dataframe creation

By reading the tabular data



Dataframe creation



```
In [2]: data = pd.read_csv("global_car_sales_data.csv")
data.head()
```

	Brand	Model	Year	Sales	Revenue	Region
0	Honda	Focus	2019	12368	20411590.85	South America
1	Toyota	Civic	2019	19893	27850924.50	Oceania
2	Audi	A4	2022	18268	49854604.49	Oceania
3	BMW	Altima	2017	14366	21423075.11	Europe
4	BMW	Jetta	2022	15679	46659156.38	South America

head(), tail() and dtype

- Head(n) and tail(n) methods show the first and last n rows of the dataframe, respectively.

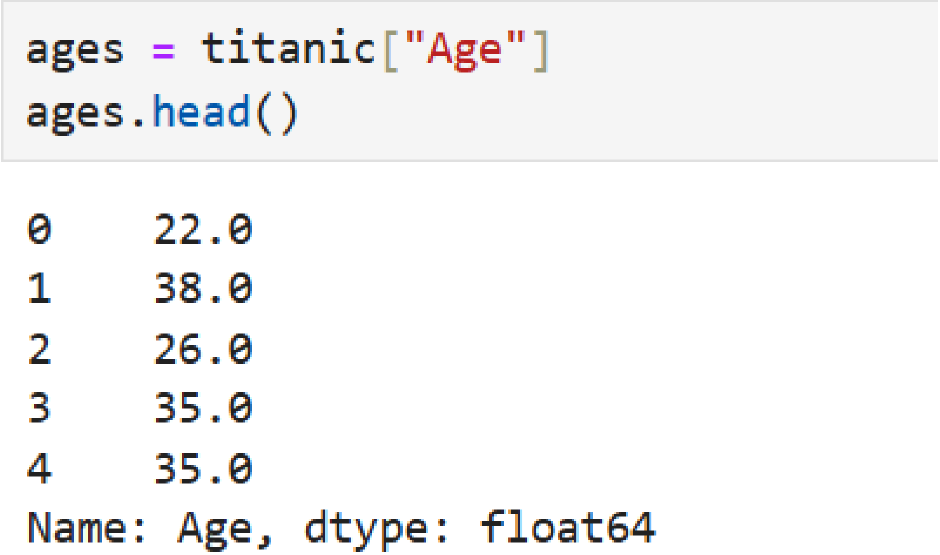
- dtype attribute shows the datatypes of each

Summarizing Dataframe

- Info():
- Describe():
- Agg()

Selecting data

- Selecting a single column: specify in selection brackets

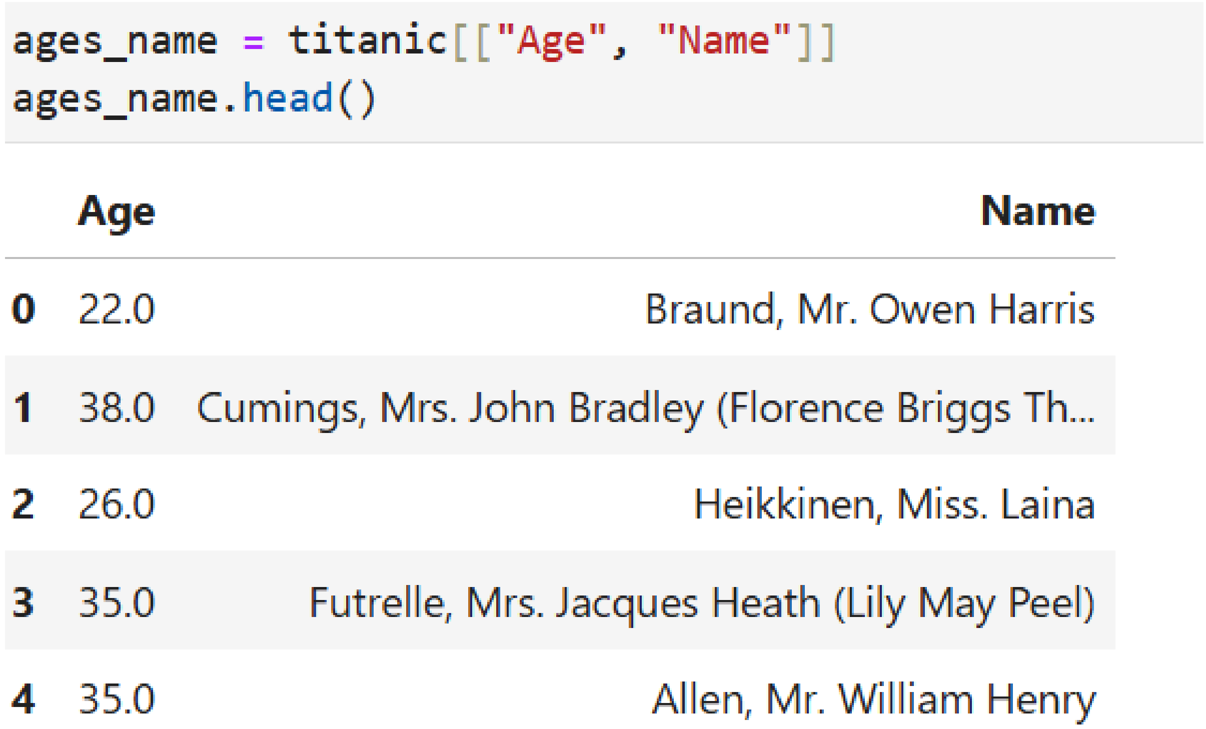


```
ages = titanic["Age"]
ages.head()
```

```
0    22.0
1    38.0
2    26.0
3    35.0
4    35.0
Name: Age, dtype: float64
```

Selecting data

- Selecting multiple columns: specify in selection brackets as a list

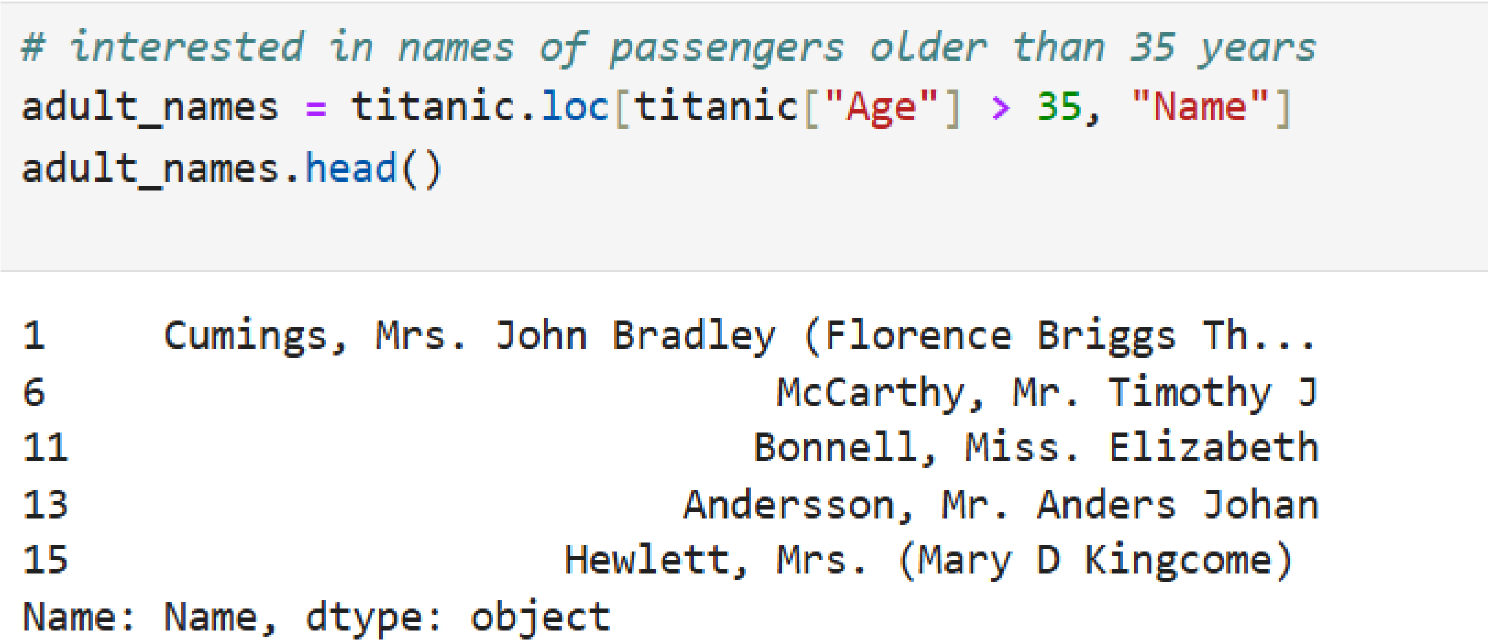


```
ages_name = titanic[["Age", "Name"]]
ages_name.head()
```

	Age	Name
0	22.0	Braund, Mr. Owen Harris
1	38.0	Cumings, Mrs. John Bradley (Florence Briggs Th...
2	26.0	Heikkinen, Miss. Laina
3	35.0	Futrelle, Mrs. Jacques Heath (Lily May Peel)
4	35.0	Allen, Mr. William Henry

Selecting data using loc and iloc

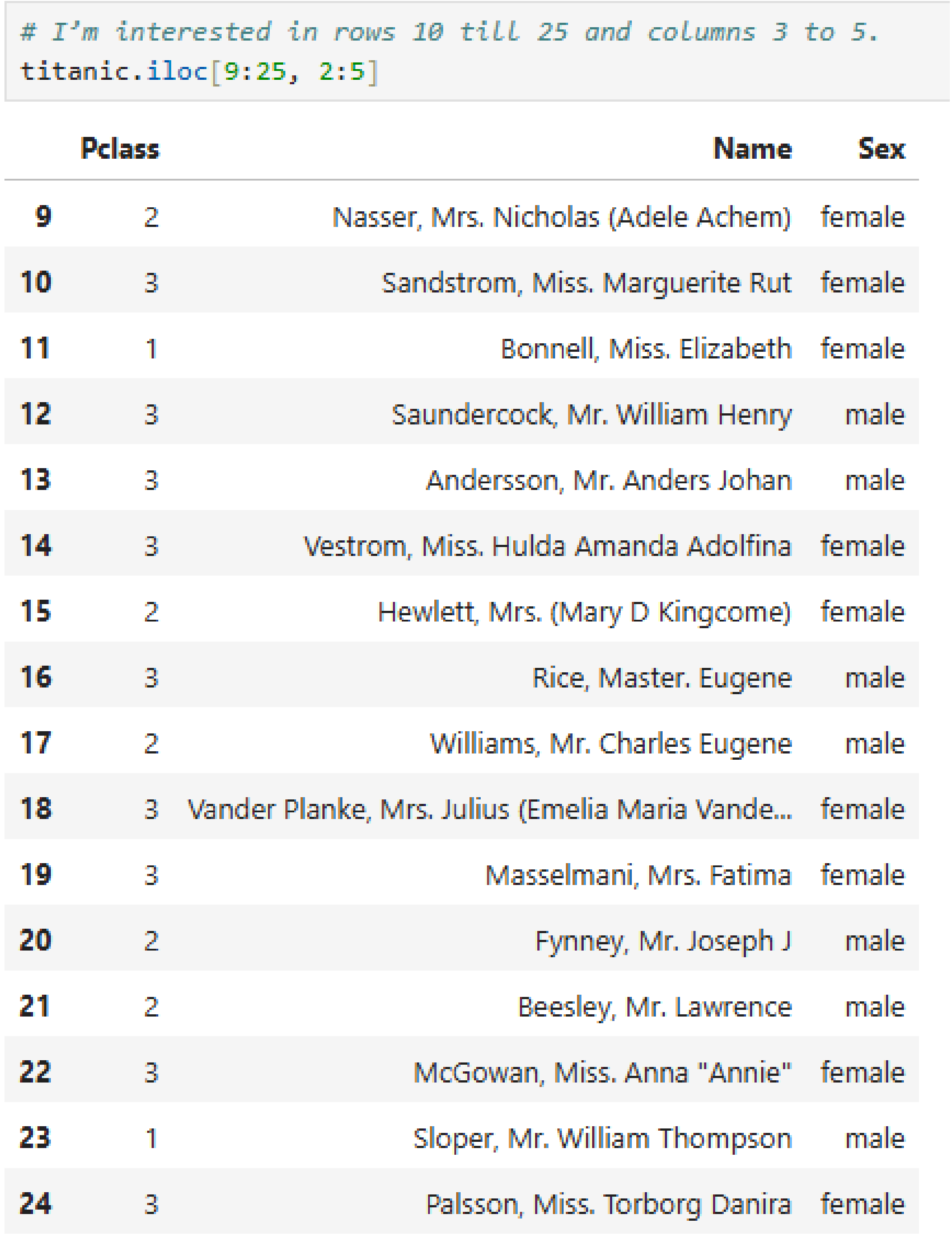
- Loc: specify names of columns:



```
# interested in names of passengers older than 35 years
adult_names = titanic.loc[titanic["Age"] > 35, "Name"]
adult_names.head()
```

```
1     Cumings, Mrs. John Bradley (Florence Briggs Th...
6                               McCarthy, Mr. Timothy J
11                             Bonnell, Miss. Elizabeth
13                          Andersson, Mr. Anders Johan
15                      Hewlett, Mrs. (Mary D Kingcome)
Name: Name, dtype: object
```

Selecting data using iloc

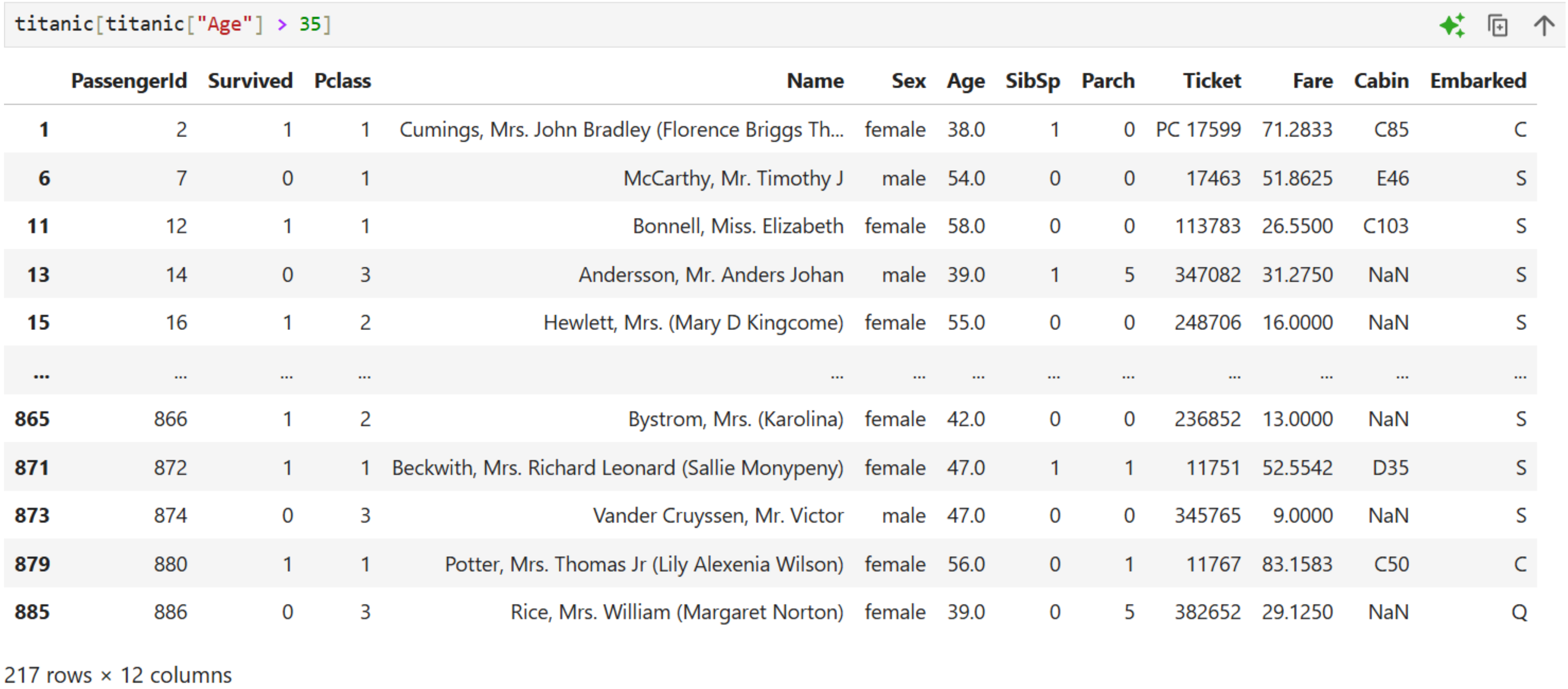


```python
# I'm interested in rows 10 till 25 and columns 3 to 5.
titanic.iloc[9:25, 2:5]
```

	Pclass	Name	Sex
9	2	Nasser, Mrs. Nicholas (Adele Achem)	female
10	3	Sandstrom, Miss. Marguerite Rut	female
11	1	Bonnell, Miss. Elizabeth	female
12	3	Saundercock, Mr. William Henry	male
13	3	Andersson, Mr. Anders Johan	male
14	3	Vestrom, Miss. Hulda Amanda Adolfina	female
15	2	Hewlett, Mrs. (Mary D Kingcome)	female
16	3	Rice, Master. Eugene	male
17	2	Williams, Mr. Charles Eugene	male
18	3	Vander Planke, Mrs. Julius (Emelia Maria Vande...	female
19	3	Masselmani, Mrs. Fatima	female
20	2	Fynney, Mr. Joseph J	male
21	2	Beesley, Mr. Lawrence	male
22	3	McGowan, Miss. Anna "Annie"	female
23	1	Sloper, Mr. William Thompson	male
24	3	Palsson, Miss. Torborg Danira	female

Filtering Data

- Using conditional expressions (all comparison and logical operators)

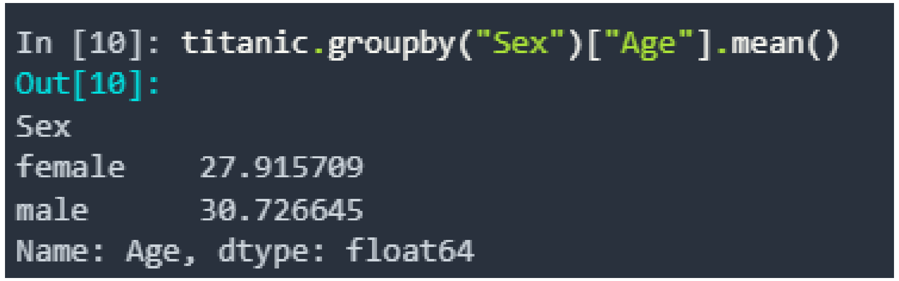


```
titanic[titanic["Age"] > 35]
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
11	12	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	113783	26.5500	C103	S
13	14	0	3	Andersson, Mr. Anders Johan	male	39.0	1	5	347082	31.2750	NaN	S
15	16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55.0	0	0	248706	16.0000	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
865	866	1	2	Bystrom, Mrs. (Karolina)	female	42.0	0	0	236852	13.0000	NaN	S
871	872	1	1	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47.0	1	1	11751	52.5542	D35	S
873	874	0	3	Vander Cruyssen, Mr. Victor	male	47.0	0	0	345765	9.0000	NaN	S
879	880	1	1	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	56.0	0	1	11767	83.1583	C50	C
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652	29.1250	NaN	Q

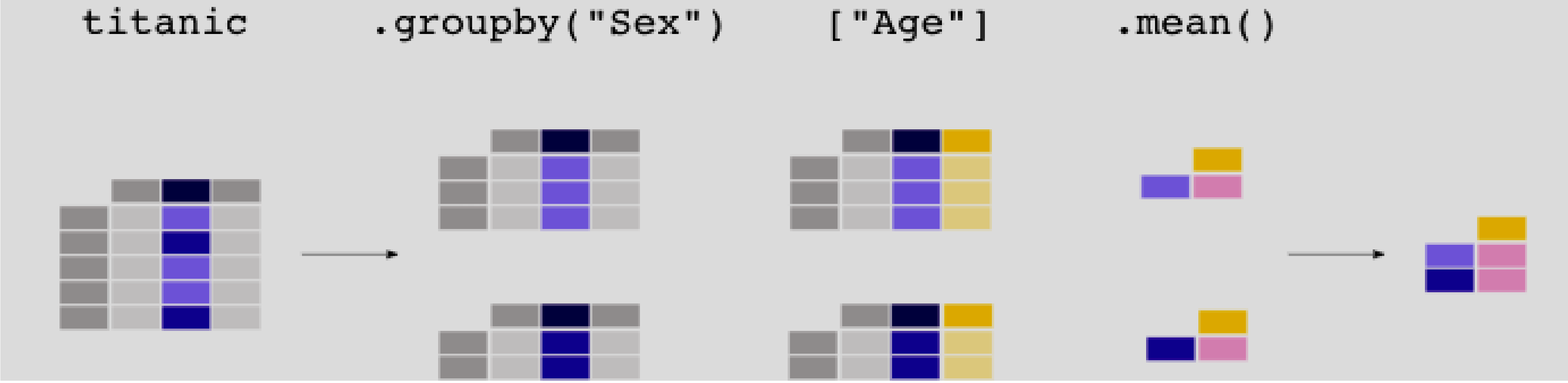
217 rows × 12 columns

Aggregating statistics grouped by category



```
In [10]: titanic.groupby("Sex")["Age"].mean()
Out[10]:
Sex
female    27.915709
male      30.726645
Name: Age, dtype: float64
```

Split the data into groups
Apply a function to each group independently
Combine the results into a data structure



Handling missing data

- np.nan represents missing data

- Remove columns that have missing data above a threshold value
- Remove rows

- Replace the missing data by some meaningful values

Excercie

- How to use the datetime

References

- https://pandas.pydata.org/docs/user_guide/10min.html#min
- https://pandas.pydata.org/docs/user_guide/cookbook.html#cookbook